

(12) UK Patent Application (19) GB (11) 2 344 963 (13) A

(43) Date of A Publication 21.06.2000

(21) Application No 9827807.0

(22) Date of Filing 17.12.1998

(71) Applicant(s)

Nortel Networks Corporation
(Incorporated in Canada - Quebec)
World Trade Center of Montreal,
380 St Antoine Street West, 8th Floor, Montreal,
Quebec, H2Y 3Y4, Canada

(72) Inventor(s)

Clive Colin Hayball
Nigel Lawrence Bragg
Niall Forbes Ross
Andrew J Tunnicliffe

(74) Agent and/or Address for Service

Franks & Co
352 Omega Court, Cemetery Road, SHEFFIELD,
S11 8FT, United Kingdom

(51) INT CL⁷

H04Q 3/00, H04L 12/24

(52) UK CL (Edition R)

H4K KFM

(56) Documents Cited

GB 2328833 A WO 98/37707 A1 WO 97/44960 A2
WO 97/22213 A1 WO 95/34974 A1

(58) Field of Search

UK CL (Edition Q) H4K KFM
INT CL⁶ H04L 12/24, H04Q 3/00
ONLINE: EPODOC, WPI

(54) Abstract Title

Object-oriented network management system

(57) A network management system for managing a communications network. The network management system's model of the network comprises a management unit level, wherein a manageable unit of network functionality is represented by a management unit model (402). Management unit models comprise a plurality of types of models, each type of model comprising classes of managed objects. The network management system's model of the network may also be viewed at a managed object level, wherein classes of managed objects may be grouped into meta-classes. Types of management unit level models include application models (405, 406), being black box representations of network functionality, and implementation models (406, 409), being white box representations of network resources.

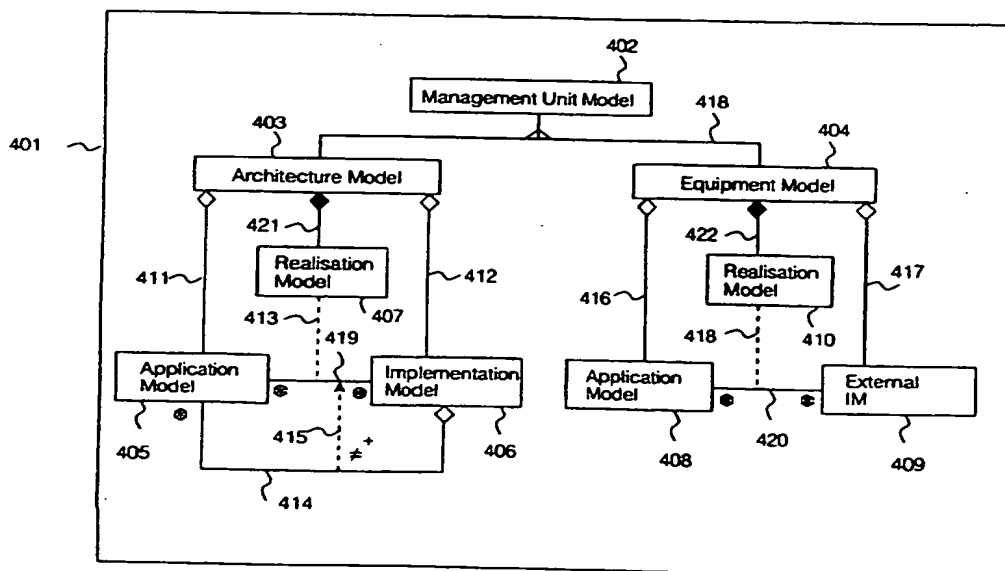


Fig.4

GB 2 344 963 A

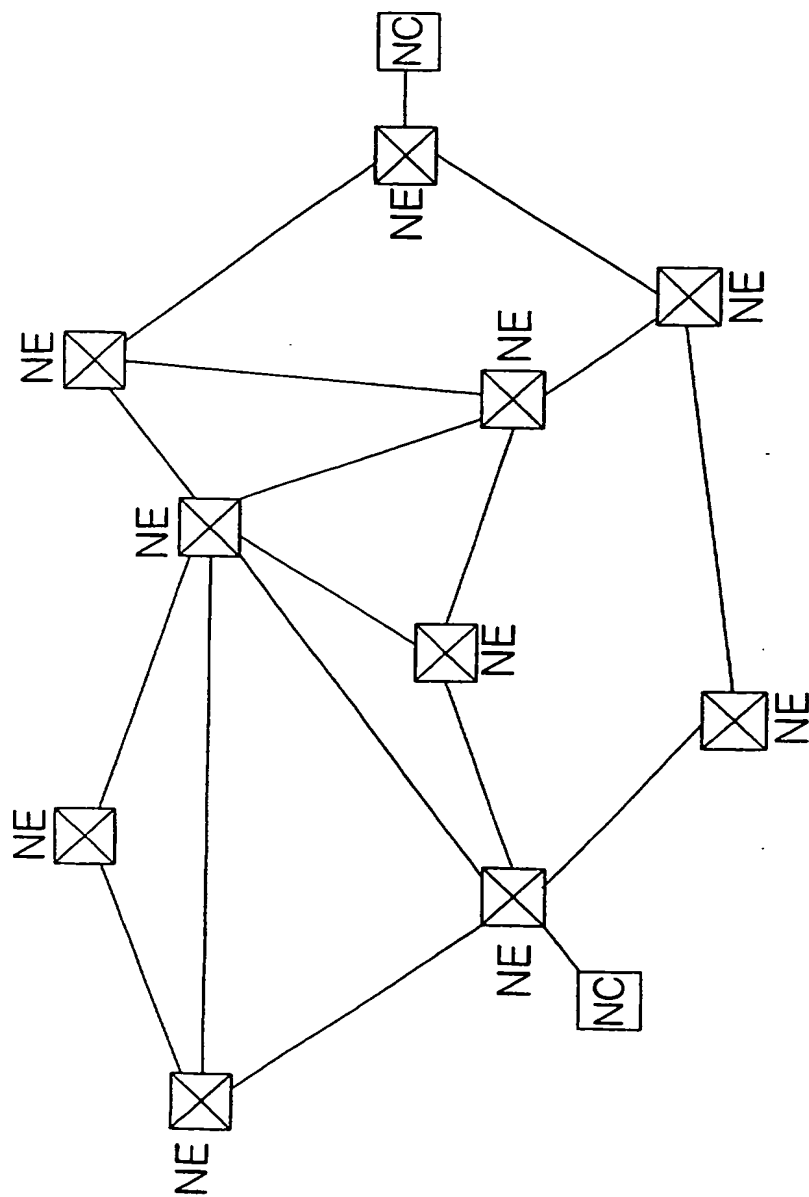


Fig. 1

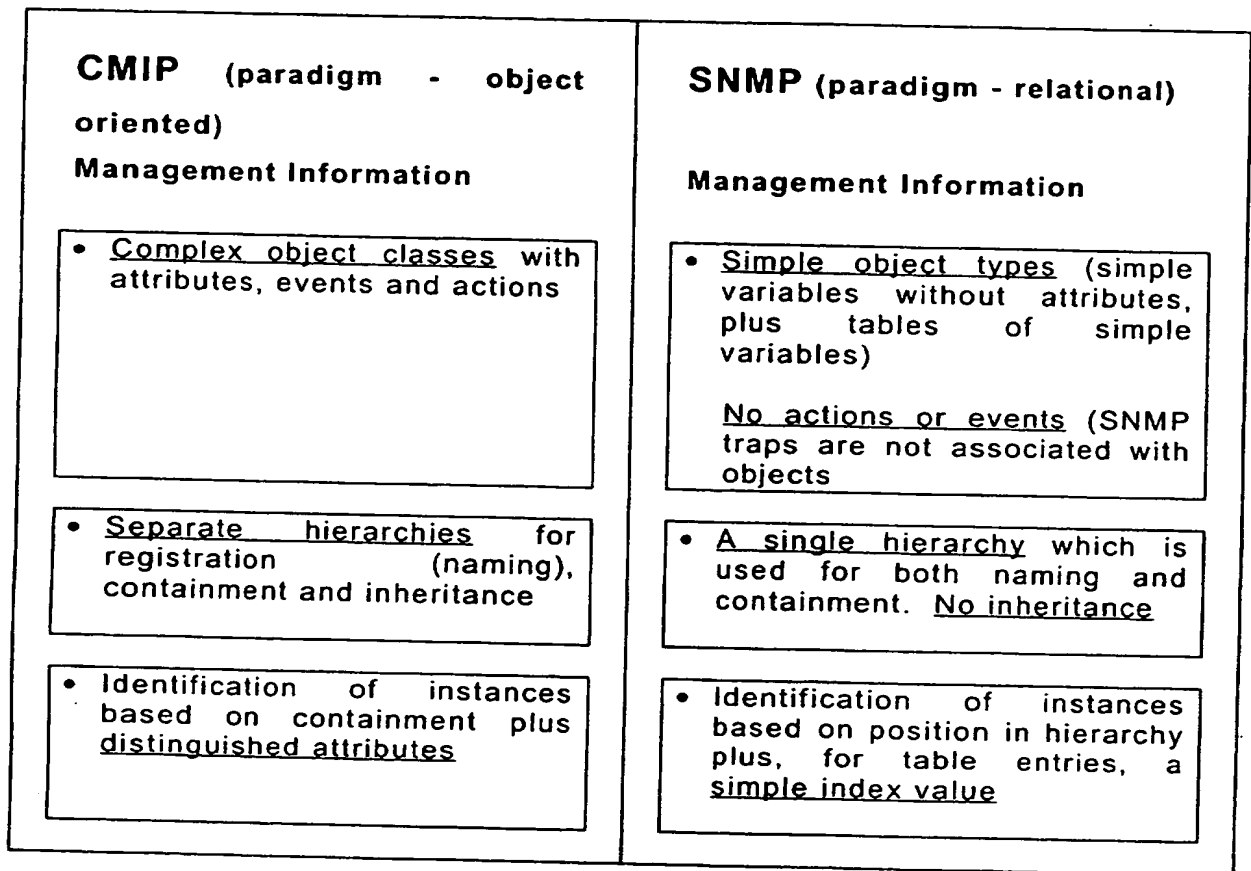


Fig. 2

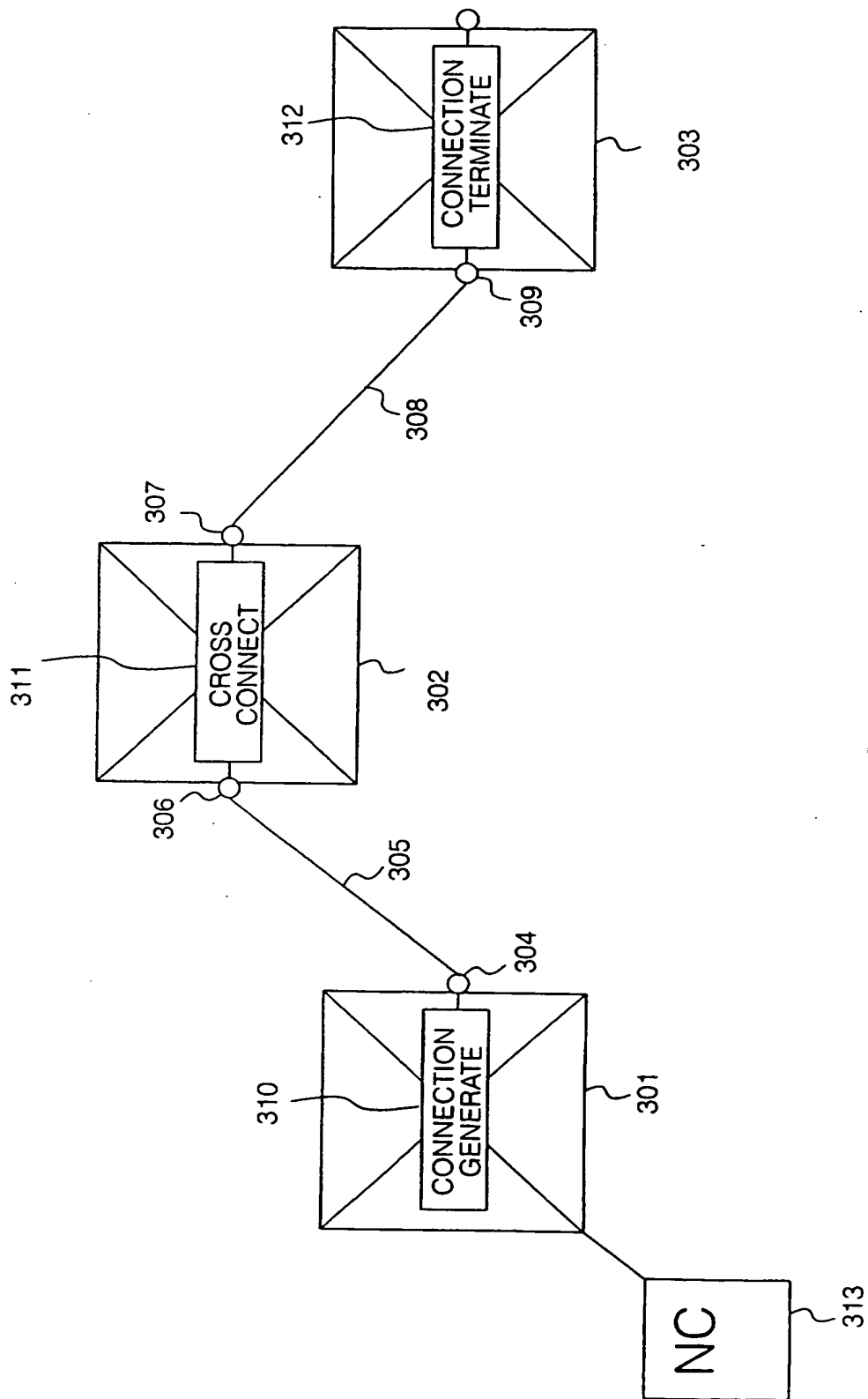


Fig. 3

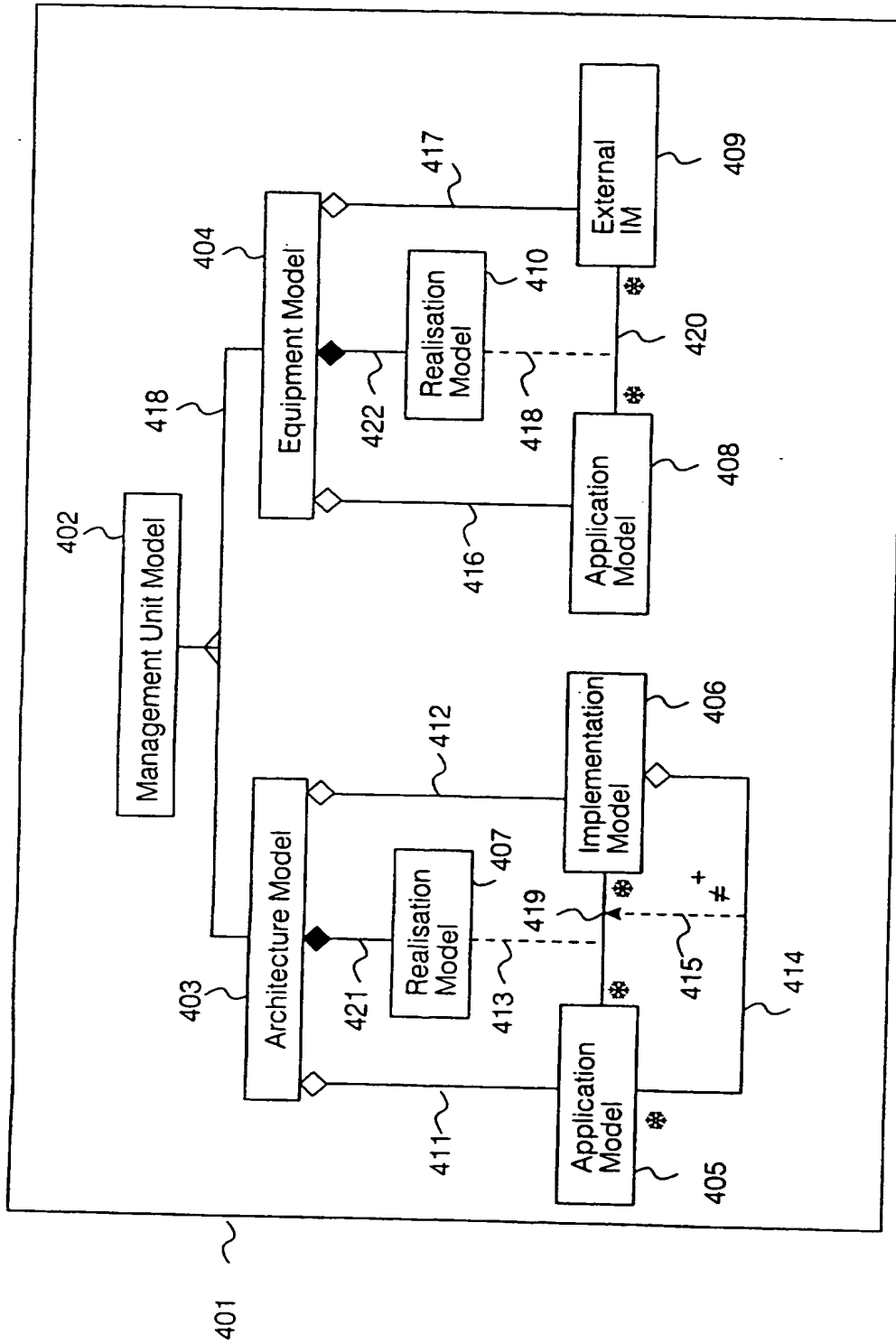


Fig.4

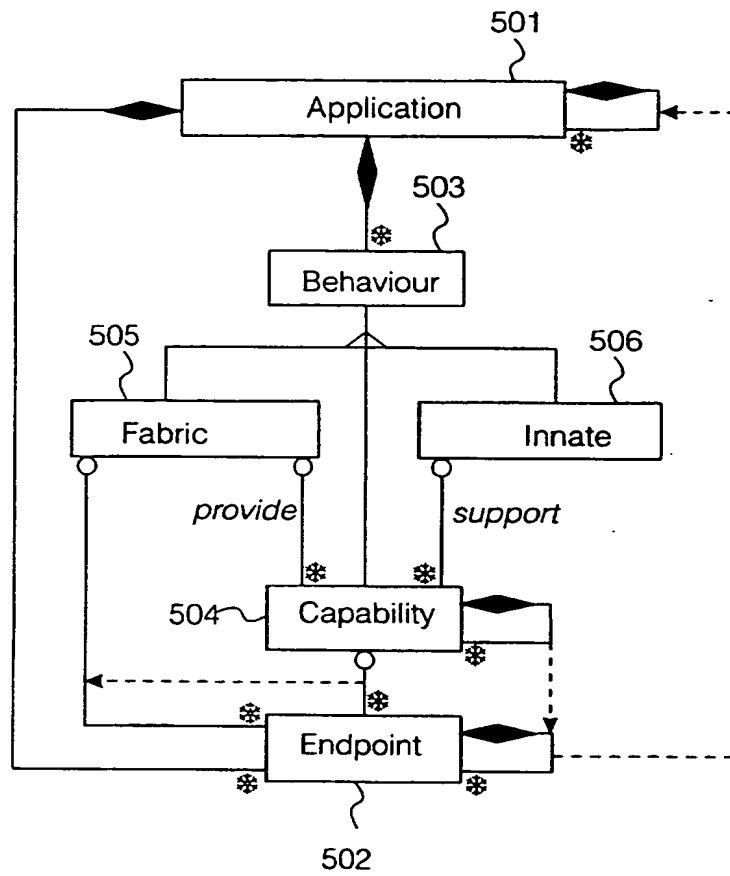


Fig. 5

SCALABLE NETWORK MANAGEMENT AND REPRESENTATION METHOD

Field of the Invention

5 The present invention relates to network management, and particularly, although not exclusively to a scalable method for representation and management of a network comprising a plurality of physical and logical resources.

Background to the Invention

10 Conventional communications networks, for example particularly broadband communications networks, result from continuous evolution of technologies over a number of past decades. A conventional communications network comprises a plurality of network elements, such as switches, cross-connects, repeaters and
15 terminals. Persons designing new communications networks, or modifications to existing communications networks have a large selection of hardware based equipment items available from different manufacturers, different equipment having different performances, and operating different transmission protocols according to different standards. Even the network equipment products available
20 from a single manufacturer can vary considerably in functionality, capability and mode of operation due to the historical development of the products, and due to takeovers, mergers and alliances between different manufacturers within the telecommunications industry. There exists a large inventory of such available "legacy" equipment, both as products currently available from manufacturers, and
25 as existing items of network equipment currently installed and in use in communications networks. On the other hand, there exists a considerable number of standards bodies and alliances of manufacturers, concerned with steering the evolution of new communications technologies and protocols with a view to standardization and interoperability of equipment from different
30 manufacturers. Many of the current standards and evolving technologies will form the legacy equipment and systems of the future, thereby adding to the volume of "legacy" equipment in use.

35 Modern broadband communications networks are required to carry an increasing volume of communications data traffic, having a mixture of traffic characteristics driven by a variety of applications. Such applications include video on demand, voice communication, and computer to computer data transfer.

P425.spec

Operators and users of networks need to be able to manage their network resources to make the most cost effective use of communications networks, and to provide adequate services to their customers. Management of a network involves management of operations, administration, maintenance, and provisioning of network resources (OAMP). Briefly, these elements involve the following:

- Operations involves the day-to-day and often minute-to-minute care and feeding of the communications network in order to ensure that it is fulfilling its designed purpose. Examples of operations include monitoring of the network by watching for faults and invoking corrective commands and/or maintenance actions to repair them, comparing measured network performance against objectives and taking corrective action and/or invoking maintenance. Taking corrective action involves operators issuing controls to correct a fault or performance problem or to resolve a customer complaint.
- Administration involves a set of activities involved with designing the network, processing orders, assigning addresses, tracking usage, and billing users of the network.
- Maintenance involves circumstances that arise when a network does not work as planned, or it is necessary to diagnose and repair system faults.
- Provisioning involves installing equipment, setting parameters, verifying that a service is operational and de-installation of equipment or services.

Conventionally, such network management may be achieved through a network controller apparatus. As illustrated schematically in Fig. 1 herein, a conventional communications network comprises a plurality of heterogeneous network elements NE, controlled and operated by one or more network controllers NC. The network elements may be of different manufacturers', and having different capabilities and protocols. A network controller NC typically comprises a workstation capable of accessing an operation, administration and management (OAM) channel communicating with each of the network elements. The network controller receives and sends messages over the operation and

management channel in the form of control signals in accordance with standard and/or proprietary protocols. Examples of widely used protocols include the simple network management protocol (SNMP) which is defined in EITF Standards. Another standard network management protocol is the known
5 common management information protocol (CMIP) of the Open System Interconnect (OSI) Committee of the International Telecommunications Union (ITU-T). Other network management protocols in use include TL1 and a variety of proprietary management protocols. Currently, the two prevailing protocols used to convey management information in communications systems are SNMP and
10 CMIP.

Although SNMP was developed with the intention of later intercepting the International Standard Organization Standards such as CMIP, there are fundamental differences between the SNMP and CMIP protocols which have
15 traditionally inhibited inter-working of the two protocols. Differences between the SNMP and CMIP protocols are illustrated in Fig. 2 herein.

Several attempts have been made to align the SNMP and CMIP protocols and models. However, many of these approaches are flawed. Such approaches
20 include:

- CMOT (Abbreviation of CMIP over ICP/IP)

The internet community have proposed how to use the CMIP
25 protocol to manage transmission and control protocol/internet protocol (TCP/IP). CMOT fails because it requires significant changes to internet equipment. CMOT has survived only as a definition of an alternative transport layer protocol for CMIP.

- IIMC (abbreviation of ISO/CCITT Internet Management Coexistence)

The Network Management Forum (NMF) has endorsed a scheme for translation between CMIP and SNMP management information
35 bases (MIBS). This approach is undesirable because it merely provides a syntactic translation between SNMP and CMIP MIB formats.

As older legacy network hardware is replaced or augmented by a new generation of network hardware which typically comprises loosely coupled elements distributed across a broadband infrastructure, the services offered by such networks need not map precisely onto a set of dedicated network elements, as the various elements may share the resources across services in a more flexible manner. This de-coupling of network services from actual network hardware presents new management challenges. As the services offered by such networks are a considerable source of revenue for network operators, management of the network services is of great importance. The network's hardware must also be managed in order to support the services effectively. Additionally, management of services needs to be coupled with management of hardware in order to automate those management processes, flow through provisioning and service impact analysis, which contribute to efficient, cost-effective operations.

The de-coupling of network services from network hardware highlights the following characteristics which can impact upon manageability of the network:

- Systems and services may be composed from logical resources, for example resilient processing platforms, diversely routed connections, which are themselves constructed out of hardware and assemblies of hardware devices. Management of such services requires manipulation of relationships between the components.
- The services provided may comprise a virtual private network (VPN) which may have a minimum quality of service (QOS) requirement for to its users. Details of underlying hardware should usually be hidden from such services.
- As the network hardware may be heterogenous, multi-technology and potentially multi-vendor and may incorporate novel and proprietary approaches to provisioning of services, there may be no standard information models to support management at the hardware level.
- Services are usually complex and may need to be managed separately from the network hardware which they utilize. Commercial requirements such as speed to market pragmatics demand reuse of resource

management functionality across different systems and services whenever possible.

ITU recommendation X.720 Management Information Model proposes a method of exchanging information at a management interface. Network resources are modeled as objects, and a management view of the resources comprises managed objects. Objects with similar attributes may be grouped into object classes. An object is characterized by its object class and object instance, and may possess multiple attribute types and associated values. The terms "managed object class" and "managed object instance" apply specifically to objects that are being managed.

Although managed objects have proved useful in aiding management of heterogenous network, a level upon which they represent network elements may still be considered to be fairly low and hardware dependent. This low level of representation means that use of managed objects can be cumbersome and require a great deal of work in order to model management of a realistic network. For example, if a piece of network hardware develops a fault, in order to continue support of a service which uses the hardware, it may be necessary to reconfigure the network controller so that managed objects associated with an alternative item of hardware which may be used to implement the same service. This reconfiguration of managed objects may have repercussions throughout the network controller's management information base, requiring further reconfiguration of managed objects associated with the original faulty hardware or effected by the replacement. A conventional management information base (MIB) represents states of managed objects and how they are connected which may not provide as much information and flexibility as is desirable for management of complex services and may not describe how high-level capabilities of the network are implemented in terms of low-level ones.

A further problem associated with use of managed objects as described in the prior art is that conventional object oriented notation which may be used to express the configurations of managed objects may be considered to be limited in some respects. Examples of such conventional object oriented notations include UML (Unified Modeling Language), OMT (Object Modeling Techniques). Weaknesses of such prior art object oriented notations include:

- their concentration on binary relationships;

- commutativity constraints on pairs of relationship meta-classes;
- constraints on larger closed loops of relationships in meta-models;
- 5 • constraints on classes and relationships occupying particular roles in context of particular models.

10 Proposed prior art solutions for improving use of managed objects are found in international patent application PCT/SE 95/00711 (Ericsson), which discloses a method of creating black box and white box models on an interface between an operations support network and a managed network, with a hierarchical composition of models based on the ITU's managed element concept. However, the disclosure of PCT/SE 95/00711 does not provide a consistent, comprehensive linked network model using both black box and white box models together in order to fully support truly integrated systems management.

Summary of the Invention

20 One aim of the present invention is to raise a semantic level of network management modeling in order to simplify and enhance network management. Such a modeling approach can support an integrated management paradigm. Specific implementations according to the present invention are intended to raise a semantic level at which communications networks are modeled in a way which is compatible with existing standards, preferably by systematically grouping low level managed object classes found in a management information base of a typical network management system into higher level units of modeling. The units of modeling and relationships between them can capture and organize information describing how high level capabilities and services provided by the network can be implemented in terms of lower level managed objects, for example managed objects representing physical resources of the network.

30 Another object of the present invention is to provide a simple, consistent approach to integrated management of distributed telecommunications networks which can provide users of a managed system with an ability to see and manage a network's elements and services from a variety of view points. Mappings between elements of a model resulting from the methods disclosed herein can support needs of individual operational practices.

According to a first aspect of the present invention there is provided a network management system for managing a network of physical and logical resources wherein said resources of said network and functionalities provided by said resources are represented by managed objects, said network management
5 system comprising:

at least one network controller apparatus, said network controller apparatus having

10 at least one data processor for processing managed object data representing said physical and logical resources of said network and managed object data representing functionalities of said physical and logical resources;

a data storage means for storing said managed object data representing
15 said physical and logical resources and said managed object data representing functionality, wherein said managed object data representations are arranged to represent said physical and logical resources and said functionality according to at least one management unit model in which classes of said managed objects are arranged into types of models, each said management unit model
20 representing a manageable unit of network functionality.

Preferably a said type of model comprises an application model comprising said classes of managed objects representing a black box view of said network
25 functionality.

Preferably a said type of model comprises an implementation model comprising said classes of managed objects representing a white box view of said network resources.

30 Said types of model comprise:

an application model comprising said classes of managed objects representing a black box view of said network functionality;

35 an implementation model comprising said classes of managed objects representing a white box view of said network resources; and

a realization model comprising associations between a said implementation model and a said application model.

Said model types preferably included an architecture model, and an
5 equipment model.

Preferably said classes of managed objects are arranged into meta-classes.

Said managed object meta-classes may be selected from the set:

10

application class;

behavior class;

fabric class;

innate class;

15

capability class;

end point class.

According to a second aspect of the present invention there is provided a
network management system for managing a communications network wherein
20 functionality and resources of said communications network are represented by
managed objects, said network management system comprising:

a management unit level wherein classes of said managed objects are
arranged into types of models, each said management unit model representing a
25 manageable unit of network functionality; and

a managed object level wherein a plurality of said managed objects are
arranged into one or more meta-classes.

30 Said management unit level and said managed object levels may be
represented by means of a substantially unified modeling language-like notation.

According to a third aspect of the present invention there is provided a
method of creating a network management system for managing a network of
35 physical and logical resources, wherein said resources of said network and
functionalities provided by said resources are represented by managed objects,
said method comprising the steps of:

creating at least one management unit model in which classes of said managed objects are arranged into types of models, each said management unit model representing a manageable unit of network functionality; and

- 5 storing said managed objects in a data storage device, said managed objects arranged according to said management unit model.

The method preferably further comprises the step of creating an application model comprising said classes of managed objects representing a black box view of said network functionality.

The method preferably further comprises the step of creating an implementation model comprising said classes of managed objects representing a white box view of said network resources.

15 The method may further comprise the steps of:

creating an application model comprising said classes of managed objects representing a black box view of said network functionality;

20 creating an implementation model comprising said classes of managed objects representing a white box view of said network resources; and

creating a realization model comprising associations between said implementation model and said application model.

Preferably the methods further comprises creation of an architecture model and an equipment model.

- 30 18. The classes of managed objects are preferably arranged into meta-classes. Said managed object meta-classes may be selected from the set:

at least one application class;
at least one behavior class;
35 at least one fabric class;
at least one innate class;
at least one capability class;
at least one endpoint class.

According to a fourth aspect of the present invention there is provided a method of construction of a network management apparatus, for managing a plurality of physical and logical resources of said network, said network management apparatus comprising at least one processor, and at least one data storage means, said method comprising the steps of:

representing a configuration of said network in data, by creating a management unit model representation representing physical and logical resources of said network and configurations of said network resources for provision of functionality, said management unit model representation comprising:

at least one architecture model, said architecture model comprising a data representation of physical and logical resources of said network which are generic to a plurality of different specific implementations of physical hardware devices; and

at least one equipment model, said equipment model comprising a data representation of specific physical and logical resources within said network; and

storing said management unit model representation data in said data storage device.

Said architecture model data representation may comprise:

at least one application model data representation, said application model data representation representing a "black box" view of functionality, provided by a plurality of physical and logical resources represented by a plurality of "white box" object models;

at least one implementation model data representation comprising a plurality of managed objects, each said managed object representing a said "white box" representation generic to a plurality of real physical devices; and

a realization model data representation, said realization model data representation describing a plurality of mechanisms by which said "white box" managed object represented in said implementation model carry out functionality

and/or services represented by said "black box" managed objects comprising said application model.

Said equipment model data representation may comprise:

5

an application model data representation, said application model data representation comprising a plurality of "black box" managed objects, each representing a functionality of one or a plurality of physical and/or logical resources of said networks;

10

an implementation model data representation, said implementation model representation comprising a plurality of "white box" managed objects, each said "white box" managed object being specific to and representing one or more individual specific devices comprising said physical and/or logical resources of said network; and

15

a realization model data representation, said realization model data representation comprising data describing a plurality of mechanisms between said "black box" managed objects of said application model representation, and said "white box" managed objects of said implementation model representation.

20

Brief Description of the Drawings

For a better understanding of the invention and to show how the same may be carried into effect, there will now be described by way of example only, specific embodiments, methods and processes according to the present invention with reference to the accompanying drawings in which:

25

Fig. 3 illustrates schematically part of a broadband communications network intended to illustrate an example of a simple service provided by the network which is representative of one of many services supported by a communications network, and which specific implementations of the present invention are aimed at representing and managing;

30

Fig. 4 illustrates schematically a management unit level model according to a specific implementation of the present invention wherein classes of managed objects are arranged into types of model, the model types each representing configurations of physical and/or logical resources within a network, which are modular and reproducible; and

35

Fig. 5 illustrates schematically set of meta-classes of managed objects according to a preferred implementation of the present invention.

5 **Detailed Description of the Best Mode for Carrying Out the Invention**

There will now be described by way of example the best mode contemplated by the inventors for carrying out the invention. In the following description numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be apparent however, to one skilled in the art, that the present invention may be practiced without limitation to these specific details. In other instances, well known methods and structures have not been described in detail so as not to unnecessarily obscure the present invention.

15 Referring to Fig. 3 herein, there is illustrated schematically an example of a typical service provision problem within a broadband communications network.

Three network nodes labeled 301, 302 and 303 are shown in Fig. 3. At the nodes are located physical resources of the network, for example switches, cross connects, multiplexers or the like. Also shown is a network controller 313 which is connected to first node 301. First node 301 comprises a communications port 304 which is linked to a communication port 306 of second node 302 by means of first link 305. Node 302 also comprises a communications port 307 which is linked to a communications port 309 of third node 303 by means of second link 308.

A simple example of a service provided by the network incorporating the components shown in Fig. 3 will now be described. The service may be managed by network controller 313 which stores data relating to management of the service and network resources and also communicates signals to the network resources in order to control implementation of the service. In the simple example described herein the service comprises providing a connection for transfer of data between first node 301 to third node 303. Devices contained within the nodes which may be utilized in order to implement the connection include connection generator means 310 in first node 301, cross connect means 311 in second node 302 and connection terminate means 312 in third node 303.

The problem is to efficiently manage a large number of services, functions and physical and logical resources comprising a broadband network as represented in one instance by the simple example of Fig. 3 herein, and to provide a management method and apparatus resident on one or more network controllers and usable by one of more network operators for managing a complex large scale communications network comprising a plurality of physical and logical resources. The network management system is created according to an architecture model presented herein, in which a plurality of physical resources and logical resources of a network, the physical and logical resources cooperating to provide network services, are represented by a plurality of managed objects arranged in accordance with a pre-determined architecture structure presented herein, and in which individual services provided by a network may be represented.

Specific network management embodiments comprise a plurality of data processors, one or a plurality of data storage means, one or a plurality of user interfaces, and one or a plurality of interfaces for communicating with the physical resources of the network, are created and configured in accordance with the architectural models represented herein which comprise specific methods for creation of such network management systems.

Functions and hardware associated with a communications network and services provided by the network are represented using managed objects. Services and functions on which services are built which are managed by a network controller in a network may be represented by "black box" objects, that is, objects which are descriptive of a function or service. Inputs and outputs of a black box object maybe known, but no knowledge of internal structure of processing of the underlying code objects, such as program instructions or physical hardware, is represented in the "black box". Managed objects representing network resources which are managed by the network controller and for which the internal structure or processing, for example specific hardware devices or algorithms, are evident in the object and are referred to herein as "white box objects".

Considering the example shown in Fig. 3, in the present specific embodiments black box managed objects are used to represent an overall function provided by a service, eg to implement a connection between first node 301 and third node 303, as well as implementing individual sub-functions required

by the overall function. For example, black box managed objects may represent a generate connection function 310 at first node 301, a cross connect function 311 at second node 302 and a connection terminate function 302 at third node 303. Examples of white box managed objects associated with the connection
5 may include managed objects representing the network resources involved in the connection, for example a node element of first node 301, port 304, network link 305, and port 306 of second node 302.

Herein, where the term "represent" is used this term includes but is not
10 limited to storing data describing one or more logical resources and/or one or more physical resources and/or one or more services and/or one or more functionalities, and the term data representation is construed accordingly. The term "represents" also encompasses configuration of at least one data processor and/or data storage device and/or signaling capabilities for communicating with
15 and/or controlling network entities which are managed or manageable.

An advantage of this separation of managed objects into black box and white box managed objects disclosed herein is that the white boxes associated with functionality of a service may be organized separately of black or white
20 managed objects associated with network resources which may be used to implement the service. Implementation of the service is described by associating particular black box objects with particular white box objects, ie specifying what functions to perform (black box objects) and which resources upon which to perform them (white box objects). This separate organization of management of
25 network functionality (eg services) and management of physical resources can yield advantages. For example, if a network resource used for a particular service becomes faulty, alternative network resources may be used instead by substituting the white box managed objects managing or representing the faulty resource with another set of white box objects associated with alternative network
30 resources which are capable of providing functionality required by the black box managed objects representing the service.

Further, sets of pre-configured objects can be stored and accumulated in managed object libraries, which promotes reuse of design effort in creating the
35 objects, and a modularisation of the network management system.

According to the preferred embodiments herein there is provided a notation for modeling a network and a method for creating an architecture model for use in

management of a network. In this specification, a "model" comprises a representation of a network, in particular data describing a network for use in management of the network possibly stored in a memory of a network controller as part of a management information base. In a preferred embodiment a model
5 contains a set of classes and relationships between the classes expressed by a notion. The model may be considered at two levels: a management unit level and a managed object level. An aim of the management unit level is to raise a semantic level of description of network management whilst maintaining compatibility with existing standards, eg standards relating to managed objects.
10 At the management unit level, the managed network is seen as at least one model, each model being one of a type defined by the modeling method, each type of model being a configuration of managed object level classes and relationships representing some manageable unit of network functionality. Each such model may thus be considered to be a "meta-class" (a class of classes) of
15 managed object classes. Furthermore, at the managed object level, the managed objects, typically found in a management information base of a management system, may be configured according to the preferred implementation into meta-classes, each metaclass being a configuration of classes of managed objects. The managed objects are preferably as defined by
20 current standards, including the CCITT recommendation X.720. The present embodiment may provide users of a managed system with the ability to see and manage a network's entities at a variety of levels - both physical and logical - with mappings between levels to ensure that the managed system's functionality can support a close fit to needs of individual operational practices. These features
25 may result in significant reduction in operational costs for network operators.

In the preferred embodiment, each type of class at the management unit level is preferably represented using object oriented notation. In the preferred
30 embodiment the object oriented notation used comprises the known unified modeling language (UML), see for example "UML Distilled", Martin Fowler, Addison Wesley, 1997, augmented by mathematical notation used to overcome certain limitations in UML's expressiveness. It will be understood by those skilled in the art that UML is merely one example of an object oriented notation and that
35 other semantically equivalent notation, for example the known object modeling technique (OMT), or Booch notation could also be used. It will further be appreciated that such notations are modeling languages used to express designs such as communications networks and whilst the specific implementations disclosed herein are described predominantly in UML, the methods and

embodiments of the invention in general are not limited by any particular notation and are notation independent. The preferred implementation intends to provide a method of creating a model using the notation, thereby being a method comprising a modeling language (notation) and steps which may be used in order
5 to create the model.

Fig. 4 of the accompanying drawings illustrates schematically a management unit level model 401 representing a management unit, according to the preferred embodiment. A single highest level (root) type of class in the
10 management unit level model 401 comprises management unit (MU) model 402.

In the best mode herein, a management unit model comprises classes (or types of models) and relationships, expressed in a substantially UML-like notation. The relationships impose constraints on the management unit model's
15 classes, such as whether an instance of a class (an instance being a particular configuration of a class' attributes representing a particular managed element in the network such as a service or physical resource, depending on the type of class) can exist without another instance related to it and how many instances can be related to another. An instance of a model is a set of class instances and
20 relationships which respect these constraints. To "instantiate" a model means to configure attributes of its classes so as to create a set of instantiated classes which do not reference any existing instantiated models. To "modify" a model means to modify one model instance into another by deleting some of its classes' instances and/or instantiating others.

25 The management unit model 402 consists of an architecture model (ArM) type 403 and an equipment model (EM) type 404. Relationships between model types at the management unit level and meta-classes at the managed object level model are denoted in Fig. 4 herein using various kinds of lines. Some of the
30 line notation used is derived from the UML notation. For example, a solid line having a white (outline) diamond at one end denotes that a model at the diamond end contains a model(s) at an other end of the line. The containment relationship denotes that contained models (at the non-diamond end of the line) are part of the containing model (at the diamond end of the line). Thus, in Fig. 4 architecture
35 model 407 contains one or more first application models 405 and one or more first implementation models 406. First implementation model 406 contains one of more first application models 405. Similarly, equipment model 404 contains one

or more second application models 408 and one or more external implementation models 409.

5 A solid line having a solid (black) diamond at one end denotes that a model at the diamond end of the line is composed of models at an other end of the line. A composition relationship, can be thought of as illustrating that the model at the black diamond end of the line is composed of the models at the other end of the line, furthermore, the models which the composed model comprise may only belong to one (whole) model, and may not exist independently of it. Thus in Fig. 10 4 architecture model 403 comprises first realization model 407. Similarly equipment model 404 comprises second realization model 410.

15 A line between two models having an asterisk at one end indicates that the relationship denoted by the line is a one-to-many relationship, with a model at the asterisk end of the line usually representing a plurality of models. A solid line between two models denotes that the two models are related in some way, for example by aggregation or composition relationships. In UML such relationship lines may be called "associations" whereas in the preferred embodiment they are restricted to denoting connections only. A broken line emerging perpendicularly 20 from a center of a line between two models may link the line to an association class. In the preferred embodiment, association classes may be called "behaviors". A behavior is preferably a class in a managed object level application model metaclass which models some data transforming behavior in a network device or function/service represented by the application model.

25 Further symbols, representing known mathematical concepts and relations (see for example "Dictionary of Computing", Oxford University Press, 1996) used in conjunction with the UML notation by the preferred embodiment may appear at ends of relationship lines are given in the table below:

30

Meaning	Symbol
reflexive (symmetric)	=
irreflexive (anti-symmetric)	≠
Transitive	→
equivalence	≡ (or ~)
partial order	≤
total order	<

reflexive transitive closure (commutative)	$=^*$
irreflexive transitive closure (anti-commutative), partially ordered closure (means the same)	\neq^* \leq^*

Commutativity restraints specified by the models may be as follows:

- 5 • for a pair of relations (between types of classes in a management unit level model), R and S the relations commute if R followed by S necessarily equals S followed by R, ie $R \circ S = S \circ R$. The relations anti-commute if R followed by S can never equal S followed by R, ie $R \circ S \neq S \circ R$.
- 10 • an involute relation is one that relates a class to itself or to one of its superordinate classes or subordinate classes. An involute relation is reflexive if it always connects each instance to itself and irreflexive if it never does.
- 15 • any pair of relations (between types of management unit level classes) R and S, for which a range of R intersects a domain of S defines an involute relation: $R \circ S \circ R^{-1} \circ S^{-1}$.

20 Management unit model 402 preferably represents a particular element (the management unit) which is managed in the network, and may be considered to be a unit of granularity in the modeled network, so distinguished for use in managing the network. For example, the management unit may comprise part of the network or an element associated with the network which may be considered to be a single, self-contained unit as far as network management is concerned,

25 for example a particular network service provided by a network operator, or a particular item of network hardware or a particular network software application. The architecture model 403 and equipment model 304 which the management unit model consists of, may each comprise an application model type (first application model 405 and second application model 408, respectively) and an

30 implementation model type (first implementation model 406 and second implementation model 409, respectively).

35 An application model may be considered to represent an external interface of a management unit level model. An application model may appear in more than one management unit level model, representing for example substantially

identical functionality in each. An application model may further be considered to be a black box view of the management unit. Subordinate objects and classes of an application model may describe functions necessary to implement a network service for example, the subordinates of the application model may comprise
5 software modules/objects or hardware device controllers which the network controller can manage in order to implement the service using particular network resources.

An implementation model may be considered to be a white box view of a
10 management unit level model. An implementation model may appear in more than one management unit model, representing, for example, the same physical resources in each. An implementation model preferably consists of configurations of either lower level application models or of classes that interface directly with network resources, for example managed objects specifically
15 associated with physical hardware. Thus, an implementation model may be considered to consist of managed objects representing particular network resources which may be utilized when implementing a service, whereas an application model may be considered to comprise managed objects representing network functionality describing what to do with or how to use the particular
20 network resources in order to implement the service.

A realization model preferably connects a management unit model's application model to an implementation model. The realization model comprises classes of realized by associations between the application model and
25 implementation model due to relationships between the classes they relate. Such constraints are valuable in promoting clean communications network design and can allow automated reasoning techniques to be applied to the models. A realization model comprises classes of realized by associations between application models and implementation models which is unique to those models'
30 management unit models. The realization model preferable comprises "realized by" relationships, representing links between managed physical network resources and managed units of managed units of networks functionality, i.e. linking attributes of classes of the model's black box view of managed objects and its white box view of managed objects. Values of attributes of the classes
35 may be used to instantiate particular attributes of the classes linked to them by means of the realized by associations. Thus, a management unit model, for example management unit model 402, comprises an application model plus an implementation model plus a realization model. The realization model shows

how attributes of classes of the application model are realized by end points and capabilities and links of the implementation model. A "realized by" association is the only class that may exist in a management unit model outside of the management unit model's application model and implementation model.

5

In the preferred embodiment, "realized by" associations may be implemented by means of pointer attributes. Classes of application models may comprise data representing pointers to access attributes of implementation models in order to associate, for example, particular physical resources (represented by the implementation model) with particular network functionality (represented by the application model) so that the functionality may be implemented. Conversely, classes of an implementation model may comprise attributes containing data representing pointer attributes to classes of the application model which the implementation model realizes. It will be appreciated by those skilled in the art, that other means of implementing realized by associations may be used, for example, by means of a managed object representing a realization model, providing an explicit representation of its realized by associations by using inheritance to associate implementation models as specializations of application models.

20

An architecture model type, for example, architecture model 403, is a management unit whose implementation model comprises a configuration of application models. The implementation model may consist of subordinate application models of other management units. Thus, an architecture model may comprise at least one subordinate model type, each of which may represent a black box view of managed functionality. For example, an architecture model may represent a computer software application, each application model within the architecture model representing a software module which may be necessary for the overall software application to function. Another example of an architecture model may be a hardware device controller, each application model within the architecture model representing individual controllers within the hardware device controller which control individual components within the hardware device.

25

30

An equipment model, such as equipment model 404, may represent managed classes and objects associated with a device or function external to the model, e.g. a management unit consisting of an equipment model is normally defined rather than computed. An application model contained by an equipment model may be associated with an externally-realized implementation model which

35

may be an item of physical hardware. The same application model may be contained within an implementation model containing subordinate application models in an architecture model. For example, an equipment model may represent a specific item of network hardware which is a node in the network, with each application model contained in the equipment model representing a function performed by a particular device within the item of node hardware. Each application model within the equipment model may be associated with an external implementation model, which may comprise defined managed objects for particular devices.

10

In Fig. 4, architecture model 403 contains (illustrated by line 411) application model 405. Architecture model 403 also contains (illustrated by line 412) implementation model 406. Implementation model 406 contains application model 405, the containment being illustrated by one-to-many containment relationship line 414. Application model 405 and implementation model 406 are associated by many-to-many association line 419. Association line 413 emerging from association line 419 illustrates that a realization model 407 associates application model 405 and implementation model 406. Architecture model 403 contains (illustrated by line 421) realization model 407.

20

Equipment model 404 contains (illustrated by line 416) application model 407. Equipment model 407 also contains (illustrated by line 417) external implementation model 409. A many-to-many association (illustrated by line 420) exists between application model 408 and external implementation model 409. Class association line 418 extends from association line 420, denoting that realization model 410 associates application model 408 and external implementation model 409. Equipment model 404 is composed of (illustrated by line 422) realization model 410.

25

Fig. 5 of the accompanying drawings illustrates an example of managed object level meta-classes which may appear in the types of management unit level models illustrated in Fig. 4. Metaclass 501 comprises a root class. In the example of Fig. 5, the managed object level meta-classes relate to an application model so root class 501 comprises an application root class. In the case where the managed object level meta-classes are associated with an implementation model, root class 501 will comprise an implementation root metaclass and where the managed object level meta-classes are associated with a management unit model, root class 501 will comprise a management unit root metaclass. A root

35

class acts as a root of a model's containment tree and preferably comprises a name which may be used to identify the model being defined. In principle, every application, implementation or management unit class can be a root of an application, implementation or management unit model, respectively. Complex application, implementation or management unit models may comprise subordinate application, implementation or management unit classes, respectively, contained within the root class. An implementation root class may contain application root classes and/or individual managed object classes. Management unit root classes are usually not shown but left implicit when drawing management unit level diagrams. Furthermore, implementation model root classes are sometimes not shown.

Managed object level meta-classes associated with application models may comprise an end point class, such as end point class 502. End points are intended to terminate connections. End points may act as interface classes between an application model and classes of other models with which the application model combines within an implementation model or a management unit model.

End point classes may be specialized to ports (representing physical end points, eg an ATM port on an actual switch), and termination points (representing logical end points, eg data representing the ATM port). Common sub-classes of termination points may include trail and connection. A trail may be defined as a transport entity which consists of an associated pair of uni-directional trails capable of simultaneously transferring information in opposite directions between their respective inputs and outputs. A uni-directional trail may be defined as a transport entity responsible for transfer of information from the input of a trail termination point to the output of a trail termination point. Integrity of the information transferred may be monitored. A uni-directional trail may be formed by combining trail termination functions and a network connection. A transport entity may be defined as an architectural component which transfers information between its inputs and outputs within a network, possibly on a specific layer of a network. An end point may be thought of as an interface class for an application model as endpoints may receive and/or transmit but may not alter data. Attributes of an end point class may be grouped as follows:

- Type attribute which may define kinds of transmissions that physical end point (eg a port) being represented can receive and/or transmit. The type may be a name of a protocol.
- 5 • Capacity attributes which may define a rate at which end points being represented can receive and/or transmit their types of transmission. Capacities may be Boolean (1 representing all, 0 representing nothing, when the end points can provide all or nothing type of transmission), a contiguous number (of items of the end point's type can provide may be arranged in a fixed sequence).
- 10 • Reliability attributes which may define reliability offered by end points being represented, either directly in terms of error rate, etc, or indirectly by recording the end point's degree of protection or similar.

15

Specializations of end point classes may have particular combinations of such attributes.

20 Relationships between end points may comprise capabilities or links. A link relationship may comprise a connection between a first end point and one or more other end points, usually denoting that data (of a certain type) may be transferred between the linked end points.

25 Managed object level classes of an application model may comprise a capability class 504. A capability may be considered to be a behavior. Behaviors of application models may be properties of (real or virtual) devices or functions represented by the application model. A behavior is preferably directly addressable by a management system using a single command (as opposed to a link which, being coordination of data in two or more managed objects, must be managed by two or more coordinated commands). Other types of behaviors may include innate and fabric behaviors, as described hereinbelow.

35 A capability class represents a connection behavior which may representation of data between a number (typically 2) of end points within an application model. Hence, in principle, a capability connects end points of different types and/or locations and/or capacities. A capability may be said to "span" end points, typically endpoints that always belong to the same application model. Attributes of a capability may be grouped as follows:

- Capacity attributes may define a maximum rate at which data can be supplied/taken from an end point providing/associated with the capability. A default assumption may be to deduce a capability's capacity from the end point's capacity with an added assumption that the behavior transforms lineally. For example, defining an adaptation capability between an ATM port of capacity X and a frame port of capacity Y would imply that a device model could, at steady rate, convert all of X to all of Y, convert half of X to half of Y, etc. A rate at which a capability can steadily transform data arriving at one end point into data transmitted from another end point, and an expansion factor between them, may be constrained by internal characteristics of a device being modeled, providing the capability with a capacity constraint over and above that deducable from the end points' capacities.
- Type attributes may define a kind of transmissions that the physical end point being modeled can receive and/or transmit. The type may be the name of a protocol.
- Reliability attributes may define a reliability offered by the end point's capability.

A capability is preferably an exportable behavior, ie one that can directly contribute to realizing a higher level capability when its application model is part of an application model of a higher level management unit. Its exportability comes from a nature of its end points; the end points must all either be conjoined to others via links in the higher level management unit's implementation model, or themselves realize end points of higher level management units' application models. Typical sub-classes of capability classes include cross-connections (connections between identical end point types) and inter-working functions (functions between dissimilar end point types).

Fabric meta-classes of managed object level classes, eg fabric class 405, provide capabilities, and may manage a set of capabilities constrained by a number of network resources. Fabric classes may have relationships to end points but are not themselves relationships between end points, their functions preferably provide capabilities that connect sub-sets of end points. Fabric classes may track total numbers or other resource limits of sub-ordinate

behaviors, record limits to the end points between which they can provide and serve as a configuration place holder for the capacity to provide these capabilities, independent of the capabilities of the capabilities' actual presence. Existence of a parameterized class of capabilities within an application model
5 implies existence of a fabric capability to provide them. Hence, it may not be necessary to represent a fabric class explicitly in a given model.

Innate meta-classes, eg innate class 506, support other behaviors directly and internally to an application model, without mediation of end points. Examples
10 of innate behaviors may include fault behaviors, performance monitoring objects and log records.

A link is preferably a connection expressed at some level as a coordination of data in two or more managed objects, usually requiring co-ordination of two or
15 more devices to set up, and may be thought of as a non-directly addressable connection. A link externally connects two endpoints that usually belong to sub-ordinate application models within an implementation model. A link is preferably managed at some level by two or more coordinated commands (as opposed to a capability which is a connection that is directly addressable by management
20 using a single command). Typically, a link is a standard one-to-one binary relationship between end points, each link preferably connects precisely one end point to precisely one other end point. Each end point can be linked to precisely one other end point. Broadcast links are one-to-many relationships. As only a link can appear in an implementation model outside any sub-ordinate application
25 model, it is usually a link that connects end points of different application models. If a management unit's application model were replaced by its implementation model within a higher level implementation model, a link would connect a realizing end point in place of the realization. There is therefore a sense in which such a link has multiple end points at either or both of its ends. Any such set of
30 multiple end points may therefore be an ordered sequence of end point realizations.

Claims:

1. A network management system for managing a network of physical and logical resources wherein said resources of said network and functionalities provided by said resources are represented by managed objects, said network
5 management system comprising:

at least one network controller apparatus, said network controller apparatus having

10 at least one data processor for processing managed object data representing said physical and logical resources of said network and managed object data representing functionalities of said physical and logical resources;

a data storage means for storing said managed object data representing
15 said physical and logical resources and said managed object data representing functionality, wherein said managed object data representations are arranged to represent said physical and logical resources and said functionality according to at least one management unit model in which classes of said managed objects are arranged into types of models, each said management unit model
20 representing a manageable unit of network functionality.

2. A network management system according to claim 1, wherein a said type of model comprises an application model comprising said classes of managed objects representing a black box view of said network functionality.
25

3. A network management system according to claim 1 or 2, wherein a type of said model comprises an implementation model comprising said classes of managed objects representing a white box view of said network resources.
30

4. A network management system according to any one of the preceding claims, wherein said types of model comprise:

an application model comprising said classes of managed objects
35 representing a black box view of said network functionality;

an implementation model comprising said classes of managed objects representing a white box view of said network resources; and

a realization model comprising associations between a said implementation model and a said application model.

5 5. A network management system according to any one of the preceding claims, wherein a said type of model comprises an architecture model.

6. A network management system according to any one of the preceding claims, wherein a said type of model comprises an equipment model.

10

7. A network management system according to any one of the preceding claims, wherein said classes of managed objects are arranged into meta-classes.

15 8. A network management system according to claim 7, wherein said managed object meta-classes are selected from the set:

application class;
behavior class;
20 fabric class;
innate class;
capability class;
end point class.

20

25 9. A network management system for managing a communications network wherein functionality and resources of said communications network are represented by managed objects, said network management system comprising:

30 a management unit level wherein classes of said managed objects are arranged into types of models, each said management unit model representing a manageable unit of network functionality; and

35 a managed object level wherein a plurality of said managed objects are arranged into one or more meta-classes.

35

10. A network management system according to claim 9, wherein said management unit level and said managed object levels are represented by means of a substantially unified modeling language-like notation.

11. A method of creating a network management system for managing a network of physical and logical resources, wherein said resources of said network and functionalities provided by said resources are represented by managed objects, said method comprising the steps of:

creating at least one management unit model in which classes of said managed objects are arranged into types of models, each said management unit model representing a manageable unit of network functionality; and

storing said managed objects in a data storage device, said managed objects arranged according to said management unit model.

12. A method according to claim 11, further comprising the step of creating an application model comprising said classes of managed objects representing a black box view of said network functionality.

13. A method according to claim 11 or 12, further comprising the step of creating an implementation model comprising said classes of managed objects representing a white box view of said network resources.

14. A method according to any one of claims 11 to 13, further comprising the steps of:

creating an application model comprising said classes of managed objects representing a black box view of said network functionality;

creating an implementation model comprising said classes of managed objects representing a white box view of said network resources; and

creating a realization model comprising associations between said implementation model and said application model.

15. A method according to any one of claims 11 to 14, further comprising the step of creating an architecture model.

16. A method according to any one of claims 11 to 15, further comprising the step of creating an equipment model.

17. A method according to any one of claims 11 to 16, wherein said classes of managed objects are arranged into meta-classes.

5 18. A method according to any one of claims 11 to 17, wherein said managed object meta-classes are selected from the set:

at least one application class;
at least one behavior class;
10 at least one fabric class;
at least one innate class;
at least one capability class;
at least one endpoint class.

15 19. A method of construction of a network management apparatus, for managing a plurality of physical and logical resources of said network, said network management apparatus comprising at least one processor, and at least one data storage means, said method comprising the steps of:

20 representing a configuration of said network in data, by creating a management unit model representation representing physical and logical resources of said network and configurations of said network resources for provision of functionality, said management unit model representation comprising:

25 at least one architecture model, said architecture model comprising a data representation of physical and logical resources of said network which are generic to a plurality of different specific implementations of physical hardware devices; and

30 at least one equipment model, said equipment model comprising a data representation of specific physical and logical resources within said network; and

35 storing said management unit model representation data in said data storage device.

20. The method as claimed in claim 19, wherein said architecture model data representation comprises:

at least one application model data representation, said application model data representation representing a "black box" view of functionality, provided by a plurality of physical and logical resources represented by a plurality of "white box" object models;

at least one implementation model data representation comprising a plurality of managed objects, each said managed object representing a said "white box" representation generic to a plurality of real physical devices; and

a realization model data representation, said realization model data representation describing a plurality of mechanisms by which said "white box" managed object represented in said implementation model carry out functionality and/or services represented by said "black box" managed objects comprising said application model.

21. The method as claimed in claim 19 or 20, wherein said equipment model data representation comprises:

an application model data representation, said application model data representation comprising a plurality of "black box" managed objects, each representing a functionality of one or a plurality of physical and/or logical resources of said networks;

an implementation model data representation, said implementation model representation comprising a plurality of "white box" managed objects, each said "white box" managed object being specific to and representing one or more individual specific devices comprising said physical and/or logical resources of said network; and

a realization model data representation, said realization model data representation comprising data describing a plurality of mechanisms between said "black box" managed objects of said application model representation, and said "white box" managed objects of said implementation model representation.



Application No: GB 9827807.0
Claims searched: 1-8, 9-10 & 11-18

Examiner: Peter Slater
Date of search: 30 June 1999

Patents Act 1977
Search Report under Section 17

Databases searched:

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:

UK CI (Ed.Q): H4K (KFM)

Int CI (Ed.6): H04L 12/24; H04Q 3/00

Other: ONLINE: EPODOC , WPI

Documents considered to be relevant:

Category	Identity of document and relevant passage	Relevant to claims
A,E	GB 2328833 A (NORTHERN TELECOM)	-
A	WO 98/37707 A1 (ERICSSON)	-
A	WO 97/44960 A2 (DSC TELECOM)	-
A	WO 97/22213 A1 (ERICSSON)	-
X	WO 95/34974 A1 (ERICSSON) - see whole document	1-3, 11-13

X Document indicating lack of novelty or inventive step
Y Document indicating lack of inventive step if combined with one or more other documents of same category.
& Member of the same patent family

A Document indicating technological background and/or state of the art.
P Document published on or after the declared priority date but before the filing date of this invention.
E Patent document published on or after, but with priority date earlier than, the filing date of this application.